

IN THE CLAIMS

Please amend the claims as follows.

For the Examiner's convenience, a list of all claims is included below.

9-24  
1. (Currently amended) A method comprising:

caching a plurality of successive type hierarchy references corresponding to a data object within the data structure of the data object; and

accessing the cached type hierarchy references at run time to perform type checking of the data object by

determining, for a query type, that is within the portion of the plurality of successive type hierarchy references, if a data object type corresponding to the data object, is the query type, and if not

comparing a depth of the data object type within the plurality of successive type hierarchy references to a depth of the query type within the plurality of successive type hierarchy references;

determining that the data object type is not the query type if the depth of the data object type is not greater than the depth of the query type; and

comparing the query type to a corresponding type hierarchy reference and determining that the data object type is the query type if the query type and the corresponding type hierarchy are equal.

10-25  
2. (Original) The method of claim 1 wherein the plurality of successive type hierarchy references are cached in a data structure of the data object.

20  
-17-27  
3. (Original) The method of claim 2 wherein the data structure is a data structure of an object oriented computer language.

12-21-  
4. (Original) The method of claim 3 wherein the object oriented computer language is selected from the list consisting of JAVA, C++, C#, and CLI.

13-22-29  
5. (Original) The method of claim 1 wherein the plurality of successive type hierarchy references comprises three successive type hierarchy references.

14-23-  
6. (Currently amended) The method of claim 1 wherein the plurality of successive type hierarchy references comprises a maximum number of successive type hierarchy references based upon required by a specific application requiring type checking of the data object.

15-  
7. (Original) The method of claim 1 wherein the plurality of successive type hierarchy references comprises a number of successive type hierarchy references, the number of successive type hierarchy references dynamically determined at run time.

16-30  
8. (Currently amended) The method of claim 1, wherein the plurality of successive type hierarchy references is a portion of a greater plurality of successive type hierarchy references, further comprising:

a) determining that the query type is not within the plurality of successive type hierarchy references;

a) ~~determining that the plurality of successive type hierarchy references is insufficient to check the type of the data object;~~

b) obtaining a highest type hierarchy reference of the plurality of successive type hierarchy references from the cache;

c) accessing a subsequent data object, the subsequent data object referenced by the highest type hierarchy reference, the subsequent data object having a subsequent portion of the greater plurality of eached successive type hierarchy references corresponding to the subsequent data object; and

d) accessing the subsequent portion of the greater plurality of eached successive type hierarchy references and determining that the query type is within the subsequent portion of the greater plurality of successive type hierarchy references; and

e) comparing a depth of the data object type within the subsequent portion of the greater plurality of successive type hierarchy references to a depth of the query type within the subsequent portion of the greater plurality of successive type hierarchy references;

d) determining that the data object type is not the query type if the depth of the data object type is not greater than the depth of the query type;

e) comparing the query type to a corresponding type hierarchy reference and determining that the data object type is the query type if the query type and the corresponding type hierarchy are equal.

e) ~~repeating operations a) though d) such that type checking of the data object is accomplished.~~

9. (Currently amended) A machine-readable medium provides executable instructions which, when executed by a processor, cause the processor to perform a method, the method comprising:

    caching a plurality of successive type hierarchy references corresponding to a data object within the data structure of the data object; and

    accessing the cached type hierarchy references at run time to perform type checking of the data object by

determining, for a query type, that is within the portion of the plurality of successive type hierarchy references, if a data object type corresponding to the data object, is the query type, and if not

comparing a depth of the data object type within the plurality of successive type hierarchy references to a depth of the query type within the plurality of successive type hierarchy references;

determining that the data object type is not the query type if the depth of the data object type is not greater than the depth of the query type; and

comparing the query type to a corresponding type hierarchy reference and determining that the data object type is the query type if the query type and the corresponding type hierarchy are equal.

10. (Original) The machine-readable medium of claim 9 wherein the plurality of successive type hierarchy references are cached in a data structure of the data object.

11. (Original) The machine-readable medium of claim 10 wherein the data structure is a data structure of an object oriented computer language.

12. (Original) The machine-readable medium of claim 11 wherein the object oriented computer language is selected from the list consisting of JAVA, C++, C#, and CLI.

13. (Original) The machine-readable medium of claim 9 wherein the plurality of successive type hierarchy references comprises three successive type hierarchy references.

14. (Currently amended) The machine-readable medium of claim 9 wherein the plurality of successive type hierarchy references comprises a maximum number of successive type hierarchy references based upon ~~required by~~ a specific application requiring type checking of the data object.

15. (Original) The machine-readable medium of claim 9 wherein the plurality of successive type hierarchy references comprises a number of successive type hierarchy references, the number of successive type hierarchy references dynamically determined at run time.

16. (Currently amended) The machine-readable medium of claim 9 wherein the method further comprises:

a) determining that the query type is not within the plurality of successive type hierarchy references;

a) ~~determining that the plurality of successive type hierarchy references is insufficient to check the type of the data object;~~

b) obtaining a highest type hierarchy reference of the plurality of successive type hierarchy references from the cache;

c) accessing a subsequent data object, the subsequent data object referenced by the highest type hierarchy reference, the subsequent data object having a subsequent portion of the greater plurality of eached successive type hierarchy references corresponding to the subsequent data object; and

d) accessing the subsequent portion of the greater plurality of eached successive type hierarchy references and determining that the query type is within the subsequent portion of the greater plurality of successive type hierarchy references; and

e) comparing a depth of the data object type within the subsequent portion of the greater plurality of successive type hierarchy references to a depth of the query type within the subsequent portion of the greater plurality of successive type hierarchy references;

d) determining that the data object type is not the query type if the depth of the data object type is not greater than the depth of the query type;

e) comparing the query type to a corresponding type hierarchy reference and determining that the data object type is the query type if the query type and the corresponding type hierarchy are equal.

e) ~~repeating operations a) though d) such that type checking of the data object is accomplished.~~

17. (Original) A machine-readable medium having stored thereon at least one data object, the at least one data object having a data structure comprising:
- a type field to specify the data object type; and
  - an identifier that identifies successive type hierarchy references of the data object.
18. (Original) The machine-readable medium of claim 17 wherein the identifier is a sub-root log to store the plurality of successive type hierarchy references of the data object.
19. (Original) The machine-readable medium of claim 17 wherein the identifier is a pointer to a sub-root log, the sub-root log to store the plurality of successive type hierarchy references of the data object.
20. (Original) The machine-readable medium of claim 17 wherein the data objects are data objects of an object oriented computer language.
21. (Original) The machine-readable medium of claim 20 wherein the object oriented computer language is selected form the list consisting of JAVA, C++, C#, and CLI.
22. (Original) The machine-readable medium of claim 17 wherein the plurality of successive type hierarchy references comprises three successive type hierarchy references.
23. (Currently amended) The machine -readable medium of claim 17 wherein the plurality of successive type hierarchy references comprises a maximum number of successive type

hierarchy references based upon required by a specific application requiring type checking of the data object.

24. (Currently amended) An apparatus comprising:

a cache memory having stored therein a plurality of successive type hierarchy references corresponding to a data object;

a main memory having stored therein instructions; and

a processor to execute the instructions such that execution of the instructions causes the processor to access the cached type hierarchy references at run time to perform type checking of the data object by

determining, for a query type, that is within the portion of the plurality of successive type hierarchy references, if a data object type corresponding to the data object, is the query type, and if not

comparing a depth of the data object type within the plurality of successive type hierarchy references to a depth of the query type within the plurality of successive type hierarchy references;

determining that the data object type is not the query type if the depth of the data object type is not greater than the depth of the query type; and

comparing the query type to a corresponding type hierarchy reference and determining that the data object type is the query type if the query type and the corresponding type hierarchy are equal.



25. (Original) The apparatus of claim 24 wherein the data object includes a data structure, the data structure storing the plurality of successive type hierarchy references.

26. (Original) The apparatus of claim 24 wherein the data object includes a data structure, the data structure storing a pointer to a sub-root log, the sub-root log storing the plurality of successive type hierarchy references.

27. (Original) The apparatus of claim 24 wherein the instructions include code of an object oriented computer language.

28. (Currently amended) The apparatus of claim 24 wherein the plurality of successive type hierarchy references comprises a maximum number of successive type hierarchy references based upon required by a specific application requiring type checking of the data object.

29. (Original) The apparatus of claim 24 wherein the plurality of successive type hierarchy references comprises a number of successive type hierarchy references, the number of successive type hierarchy references dynamically determined at run time.

30. (Currently amended) The apparatus of claim 24 wherein the main memory stores further instructions such that execution of the further instructions causes the processor to:

a) determining that the query type is not within the plurality of successive type hierarchy references;

a) ~~determining that the plurality of successive type hierarchy references is insufficient to check the type of the data object;~~

b) obtaining a highest type hierarchy reference of the plurality of successive type hierarchy references from the cache;

c) accessing a subsequent data object, the subsequent data object referenced by the highest type hierarchy reference, the subsequent data object having a subsequent portion of the greater plurality of eached successive type hierarchy references corresponding to the subsequent data object; and

d) accessing the subsequent portion of the greater plurality of ~~eached~~ successive type hierarchy references and determining that the query type is within the subsequent portion of the greater plurality of successive type hierarchy references; and

e) comparing a depth of the data object type within the subsequent portion of the greater plurality of successive type hierarchy references to a depth of the query type within the subsequent portion of the greater plurality of successive type hierarchy references;

d) determining that the data object type is not the query type if the depth of the data object type is not greater than the depth of the query type;

e) comparing the query type to a corresponding type hierarchy reference and determining that the data object type is the query type if the query type and the corresponding type hierarchy are equal.

e) ~~repeating operations a) though d) such that type checking of the data object is accomplished.~~